

建筑学背景的软件互联网创业者的投资风险分析

2022 年 1 月 27 日 1.0 版本

摘要:

工程师型的建筑院校的毕业生去软件和行业互联网创业，面临三个问题：1.达不到规模效益的门槛。2.不知道自己的技术能力可能在行业中落后。3.无法获得技术。所以，投资给他们，有巨大风险。

正文:

感谢尊敬的投资部门领导百忙之中听取汇报，请各位领导多多指教。

2010 年以来，室内和建筑相关软件行业，出现了一批建筑学背景的创业人。

投资人看好他们的理由有 3 点：

1. 独立技术公司摊薄成本。独立于各个地产公司和设计院的科技公司，可以服务所有用户。
2. 建筑学背景创业者的创新理念和商业模式，有很大价值，能够变成产品。
3. 建筑学背景的创业者即使没有技术，可以通过人员的流动、技术交流和购买技术 3 种方式获得。

非常不幸，2010 年以来的几次行业内软件和互联网创业的经验表明，上述 3 条对建筑学出身的软件创业者往往都不能成立。建筑学人软件互联网创业，虽然提供了极好的新理念和新的商业模式，带活了行业，但是，创业结果往往不好，要么清盘出局，要么苦苦支撑。

要解释原因，必须从两个方面：

1. 成本-收益的逻辑和关键节点。
2. 不同建筑院校的不同策略。

分析成本-建筑学院策略-相应意识形态等方面，分析其中的内在脉络。再分析创业者面临的问题。

1. 成本-收益的逻辑和关键节点

我们知道，要进行计算机相关研究，培养一个程序员，数据科学家，人工智能科学家，都需要长期的投入。需要摊薄成本。要形成，人才、项目、资金三者的良性循环。不同类型的建筑学院有不同的应对策略。

根据国内历史学家的研究，大概在 2000 年左右，国外建筑学院的计算机辅助建筑设计研究就基本上脱离了行业的主流。国内大概在 2010 年左右，工程师院校的计算机辅助建筑设计（CAAD）研究也脱离了主流。西方和中国港台地区比国内提前 20 年左右。关键阶段如下：

1.1 成立建筑软件公司

在 80 和 90 年代，各个设计院成立了电算站，电算站开发过大量建筑用软件，由于市场上没

有专业的软件，大量的软件由设计人员在业余开发。各地电算站开过几次联合会议，交流。1985-1995 之间，国内成立了一批的软件公司。有了软件公司以后，个人的产品比不过软件公司职业化的产品。软件需要培训服务、售后服务、不断升级服务。建筑学院 CAAD 老师通过编程赚钱的机会渺茫。

1.2 科研团队逐渐分化

2000 年左右，国内的不同背景的学术团体逐渐分化。参与建筑科研课题竞争的，不仅有建筑学院，还有软件学院和土木学院等。建筑学院的老师们并不占有优势。由于课题的设立、分配中，起到决定性作用的是政府的科研院所和大型公司的“学术山头”。绝大多数课题，需要比较大的学术团队才能做。建筑学院教授计算机辅助建筑设计的老师顶多 2~3 个，形不成合力，也就拿不到国家的自然科学基金。比方说，BIM 的课题很多，建筑设计专业比结构专业更懂得 BIM；而国内高校内，结构专业做 BIM 明显比建筑专业更专业一些，并拿到更多的资源。建筑学院的教师很难拿到相应的课题。

1.3 软件公司上市

2010 年左右，国内的建筑软件公司上市，简单的功能，也需要和现有的成套软件相配合。上市的软件公司在这方面形成了巨大的优势。软件的兼容性很难做，好不容易兼容了，没过多长时间，上游软件故意升级不兼容，下游软件就非常麻烦。

建筑学院的 CAAD 教师，不约而同选择技术含量最高的项目，比如日照软件等，希望通过高技术，突破兼容性问题，并获得技术溢价，很不幸，实践结果是：即使获得了一定的技术优势，市场效果也并不好。

1.4 软件公司相互收购形成双寡头格局

2020 年左右，国内大型建筑上市软件公司收购其它软件公司，软件行业形成了双寡头局面。值得注意的是：龙头公司收购一家 200 人左右的软件公司及其大量的软件成果，价格很低，很可能是创业公司开发相应技术的成本的 1/10 左右，定价标准是什么呢？价格接近于收购公司开发收购软件的价格。

建筑学院的老师开创业公司，很难把软件的开发成本降低到收购价格。

1.5 程序员集中在一二线城市

程序员的培养成本很高，大多数小公司只能依托于一线中心城市的“程序员池”，形成多家公司共同培养程序员，共同使用的模式。2000 年以后，高水平的程序员集中在某些一二线城市，在一些二线城市没有程序员的地区，就没有办法获得高水平的程序员。如果公司规模不大，去二线城市，可能会面对没有高水平人才的窘境。2018 年，委托某个二线城市的建筑软件公司开发构件库，软件公司的开发系统中竟然用 ORACLE，会引来法律风险，让人吃惊。一线城市的软件公司几乎不可能出现此问题。

处于二线城市的建筑学院，很难获得足够的高水平程序员。

在一线城市，雇佣高水平程序员必须加上一线城市高昂的买房租房成本。如果一个项目没有万分的把握，很难启动。

想启动软件项目，大约有 3 个办法：1.抵押自己的房产获得风投，雇佣程序员。2.卖掉自己的房产，雇佣程序员。3.自己花多年时间学习，成为“全栈工程师”。

2 不同类型的建筑学院的不同策略

建筑学院内的计算机辅助建筑设计要保持水平，必须在人才、项目、资金三方面达到良性循环。而良性循环的门槛不断提高。

国内外的建筑学院可以分为两种类型：领袖型与工程师型。面对巨大的沉没成本，不同类型的建筑院校采取了不同策略。领袖型的目标是培养“专业帅才”，必须保证学生具有一定的知识量去判断，会开设难度极大的技术性课程。工程师院校目标是培养建筑师，对人工智能理论和软件开发没有太高的要求，为了减低总体成本，发展了西方中心主义的神话。。

2.1 高昂沉没成本

在建筑学院很难形成分工的情况下，进行研发或者科研，需要一个人投入将近 30 年的时间：

1. 花 5-8 年时间在高校学习建筑。
2. 3 年时间建筑设计实践。
3. 4 年脱产时间学会计算机或者数学的本科。
4. 3 年时间脱产学习数学或者计算机的硕士课程内容。
5. 为了能独立开发或者科研，起码花 10 年时间掌握编程或者实践数学，一般在软件公司上班。

建筑学院 CAAD 课程老师太少，难以分工，一般来说，要做出一个演示版本算法；必须要掌握至少一种编译型语言和一种解释性语言。还要熟悉人工智能、软件开发、服务器配置和管理等多方面的技能。全部的学习内容，要花 10-20 年，由于现代社会的同行和家庭的各种要求，很难有机会长期学习。成本极为高昂。几乎就要赌上一个人的半辈子。

2.2 领袖院校：必须达到一定的知识量，以便决策

领袖型院校的教师有很大的资源维持一个可以判断优劣的学术体系，因领袖型院校的教师往往掌握大量资源，大量资源一定面对大量的竞争，为了能在竞争中保持优势，就不得不维持一定的学术水准；也就能判断优劣，从而促进获得资源，形成良性循环。

老师有自己可以销售的软件，就会要求学生也具有写软件的能力。

领袖型的院校，会给学生更多的选择，并能提供 3 个支持：

1. 榜样。领袖型的院校往往有成功的教师，能独立开发软件或者是能拿到大量的课题，能给学生提供成功的榜样。所以我们发现，领袖院校不一定投入很多钱，就能做出比较好的成果。
2. 学术体系。领袖型院校往往有自己的学术体系，有自己的学术领袖，学生不会放弃研究计算机技术，转设计。
3. 在课程设置上，重视基础。并且校方能够坚持基础。

领袖型的院校对建筑学学生开设 C++ 编程语言课程，课程在半个学期内完成，教学速度超过

整个大学所有院系的 C++ 课程，难度极大。学生们选举了代表，向开课的建筑学院的老师和学院反映课程太难，学生们认为全国的建筑学院都没开设 C++ 课程，该课程没有用处。学院方面研究后认为，课程可能没有直接用处，学生以后可能做行业领袖，而领袖必须具备大范围的知识。“学不会”也是一种学习。

从后面的实践来看，“学不会”比其它院校的“能学会”好得多。领袖院校的学生创业以后，往往能正确评估难度，至少能做到全身而退，不会出现卖房子且出局的悲惨局面。

其中可能包含学生培养目标的问题。

2.3 工程师院校：避免高投入

工程师院校的学生和教师，很难获得自信和资源去长期学习、实践。进行软件开发或者进行科研的成本越来越高，已经到了建筑学院里老师所能承受的极限，最重要的问题并不是如何取得软件开发项目，或科研项目，保住最重要的资源——“教职”，要同时满足 3 个条件：

1. 成本低。
2. 能够产生一定的效果和影响力。
3. “区隔性”，防止软件或者结构等领域的学者正面竞争。

多年前，工程师院校也有极为优秀的老师，但是也不能阻止新来者重新解释学术，因为教师在工程师院校很难获得维持学术性的资源。由于教师不掌握大量资源，也就难以有资源维护学术等级体系。没有资源，就无法调动学生。学生们爱干什么就干什么，学生搞一些无法判断的东西。所以在工程师院校，各种所谓的“创新”层出不穷。

有两个方面可以符合上述要求，分别给高校和学生带来问题，甚至是严重的灾难：

1. 买大量的非通用的设备。

开发专用设备，需要极为高超的知识，要求对非通用硬件有相当的了解。难度极高。1990 年代到现在，海峡两岸的工程师学校，都买过巨额的非通用设备。结果都不太好。

2. 引进国外某个大学的“中间态”技术。

所谓“中间态”是说，刻意加上了计算机和建筑学的“混合”知识，让两边的人都比较难以直接评价。但是“中间态”技术，一旦有一点市场前景，立刻被数学家和软件工程师看破。因为建筑学所有知识并没有超出人类的认知。建筑学的基本问题与其他行业的知识并没有本质的不同。

工程师学校院校背景的建筑学院内，从国外引进成本不高的学术，大体上是亚历山大的《建筑模式语言》、《建筑的永恒之道》的变形。强调其中建筑的特性，自认为可以避免和其他专业，比如说计算机专业，数学专业等专业直接面对面硬碰。以此降低了学术的成本。教师不用花上 28 年时间，掌握建筑、计算机、人工智能、相关数学和计算机图形学的知识。

3 创业者面临的问题

建筑学背景的创业者，给行业带来了新的商业模式，新的理念，具体实施的时候，会遇到产业问题、没有技术的问题和衍生出一系列问题：

1. 难进入规模效应的门槛，独立技术公司相比于地产公司定制产品，可以摊薄成本，但是于行业双寡头软件公司相比，还是处于严重的劣势地位。
2. 创新理念和商业模式有很大价值。反而成为围猎的目标。

3. 长期积累的开发经验不足，难以快速开发产品。

建筑学背景的创业者难以通过人员的流动、技术交流和购买技术 3 种方式获得技术。但是会遇到相应的几个问题：

4. 难以找到技术人员合伙
5. 没有技术情报活动。
6. 市场上收购技术的报价高，报价基于购买方的开发成本。
7. B 端和 C 端的客户开发成本高于 VC 端。

3.1 难进入规模效应的门槛

独立公司可以服务社会上所有的企业，可以摊薄研发成本；与企业内的定制业务相比，有更好的成本优势。建筑软件行业并不大，为了摊薄成本，也正在走向双寡头化。

独立技术公司与双寡头竞争，难以摊薄成本，比如，双寡头公司的推销员去了客户公司，可以介绍很多款软件，而独立公司只能介绍一两款软件，独立公司的销售成本远远大于传统寡头公司。

所以，独立公司的软件产品必须是独特的，有很高的技术含量，以至于寡头行业软件公司无法模仿，或者，该产品有很强的互联网属性，一旦确立了地位，后来者会花费很大的成本才能赶超。

巨型的传统软件公司是情报活动的主要赢家。所以，他们往往不愿意砸钱初始项目，技术水平高，天下英雄，尽在掌握中，早一点，晚一点而已，好比孙猴子跑不出如来佛的手掌心。而国内的双寡头都已经投入超过了 20 年以上，新来者希望十分渺茫，除非有全新的商业模式。

3.2 创新理念和商业模式有很大价值，但成为围猎目标

技术持有者为什么不愿意与建筑学创业者合作？反而合伙围猎建筑学创业者？

我们不妨从持有技术者的角度来看问题。持有技术者认为：建筑学创业者的主要创新是社会公共财富，可以不花任何代价获得：

1. 建筑学的创业者，在销售模式，生产模式，产业模式进行了创新。哪怕这些创新起了决定性作用，建立了新的产业。这些创新由于不带有技术因素，创新非常容易被抄袭。
2. 创业者为了获得声望获得投资人的青睐，不得不对外大量的宣扬发明和创新。所有人都可以从公开渠道获得这些创新。
3. 有相关技术的个人和机构不会因为这些社会公共财富而为此自己把自己的技术卖掉，技术持有者完全可以抄袭这些社会公共财富，几乎一分钱不付就获得了建筑学的创业者的全部成果，再结合自己的技术，自己创业。而不必入股，或者销售自己的技术。
4. 2020 年以后大多数软件业人士认为，风投更像债务，而不是营收；获得风投往往需要稀释自己的股份，或者拿出自己的房产与投资人对赌，大多数软件业人士愿意自己长期持有一个软件或者软件技术，获得长期的现金流，而不愿意持有债务或者拿自己的房产冒险。软件很难预测，与其为了一个宏伟的目标而损失眼前自己的房产，还不如干脆去大公司打工。
5. 如果建筑创业者的公司给股票换技术，那相当于是用公司的债务换取了技术持有者的技术。如果技术持有者主动上门推销自己的独门技术，在谈判的过程中就会透露很大的信息，完全无法避免建筑学创业者仿制。原子弹爆炸以后，就是对仿制原子弹的极为有利的提示。

2010 年，某建筑学创业公司在商业模式上进行了创新，同时有两个创业团队有建筑学创业公司所需要的技术，其中一个技术的创业团队并没有直接开始创业，而是作为员工到了建筑学创业公司中去工作了三个月，了解了它的商业模式，然后开始了自己的创业。另外一个持有关键技术的创业团队也没有向建筑学创业公司销售自己的技术，后续创业也成功了。该“骚操作”成功以后，技术持有者都知道：技术是自己的，而商业模式是可以学习的公共财富，不会把自己的私有财富去交换社会公共财富，况且，试错商业模式需要大量资金，而商业模式又是社会公共财富，在 TO B 市场上比较难以积累客户。创业公司搞得不好，只会积累债务。技术人员不会拿私有技术财富去交换一堆债务。

3.3 开发经验不足，难以快速开发产品

开发需要技术和长期的经验，建筑学创业者缺乏相应的培训和技术：

1. 工程师建筑学院往往有“西方中心主义”神话，“西方中心主义”并不是西方世界的原创，理由在于：西方世界的建筑学院的教授们知道软件行业的水平，并没有赌上房产去创业，而是老老实实教书。“西方中心主义”神话是中国的一些工程师院校为了减低学术成本，创造的新观念。绝大多数工程师院校的学生并没有建筑历史考古学的能力，不知道计算机科学和数学等学科已经大幅发展。学生们陷入神话难以自拔。

西方中心主义的神话产生了两个效果：一方面老师轻松了，不教编程等硬核科技了。学生缺乏技术上的训练。另一方面，学生拿着国内外工程师学校学习的落后内容，抵押房产去软件行业创业，好比训练不足的士兵，拿着一战世界大战的栓动式步枪，上现代战场。

很难快速开发产品原型。

2. 由于没有相应的经验，很难进入正规的软件或者互联网公司，没有机会从软件公司获得技术经验。开发流程，技术人员之间的协调是一门高深的艺术，只能通过长期的实践获得。即使开了公司也不能获得

3. 由于工程师院校的毕业生没有相应的技术能力，很难去行业软件公司打工。缺乏对行业的了解，很难找到快速开发的人员。

3.4 难以找到技术人员合伙

投资人普遍认为，有技术能力的大牛，不愿意在大公司受到各种人事制度的羁绊，愿意自己开公司。反过来说也成立，愿意自己出来开公司的，就是有技术能力的大牛。

1. 找到和分辨谁持有关键技术，难度很大，超出了建筑学背景的人的能力。

2. 创业对普通技术人员没有吸引力。在大多数技术人员看来，获得创业投资，不是资产，而是债务。在座的诸位都是风投行业的大佬，你们有多种方式可以限制，甚至是控制创业者。而程序员多多少少也知道一些诸位的霹雳手段。程序员不愿意冒险，而愿意拿高薪。

3. 技术人员不愿意在外行手下干活。事半功倍。

3.5 没有技术情报活动

技术情报活动广泛存在。大体可以分为 3 类：

1. 公司之间，价格比较高，不讨论。

2. 公司与公司外的个人，往往用高薪为诱饵，对行业里面的前几名开发者“技术面试”，一般都可以获得比较好的效果。

3. 技术人员之间的技术情报活动。最广泛存在的技术情报活动。

技术人员直接的技术交流的一种模式如下：

1. 善意接触。往往从提交 bug 开始。
2. 评估双方对技术的深度。技术人员在私下交流技术的时候遵循一个模式，等价交换，如果对一项技术双方进行有着同等的程度，那么就可以一分钱不花进行技术交换，同时同等提高双方的技术程度。如果一方技术比另一方面技术强很多，这种技术交换就不会存在。双方尽力在对方不能获得任何知识的情况下，让对方相信，自己掌握的技术，可以解决对方的问题。也就是数学上的“零知识证明”。
3. 交换了相应的技术以后，双方往往构成了联盟关系。因为如果双方翻脸，彼此的技术重叠，会出现“你卖一块，我卖八毛”的竞争结果，对双方都不利。

如果小技术团队要完成比较大的产品。技术情报活动是个重要的手段。大家在底下偷偷摸摸交流技术，好像海底火山口附近煮沸的水。海面上很平静。工程师院校的建筑学创业者由于没有技术背景，没有潜入海底，也就不知道海底发生的事情，一点也没有参与。

3.6 购买技术，很高的报价

外行的创业者和投资人，在技术市场上收购技术，往往找不到相应的技术，或者价格高的离谱。

1. 技术人员串通

在技术市场上找不到存在的技术。从创业者角度看，技术人员明显处于囚徒困境中：技术一般在团体环境中开发，往往很多人持有全部或者部分，根据一部分技术，可以猜测出全部技术的全貌。A 技术人员不卖给创业者，创业者找到了 B，A 的技术就没用清零。所以，创业者认为自己能用很低的价格买到技术。

但 2012 年以来的非技术出身的建筑学人的几次创业，出现一个共同的情况：没有技术的创业者在市场上买不到技术，原因是技术人员串通，或多或少进行了封锁。

技术人员与创业者的博弈，天平渐渐的向技术人员倾斜，因为技术人员以前所缺乏的是三样：商业模式，资金关系，技术人员彼此之间的联系。商业模式可以模仿。风投行业发展，获得资金越来越流畅。技术人员之间的联系，用 QQ、微信、网站等工具得到了加强。技术人员在整个的博弈中占了上风往往是创业者花了很大的精力去创新的。新的模式，新的产业链被软件开发人员，数据科学家和人工智能用很小的代价就模仿了。

把债务留给创业者。

2. 自身开发成本高，获得的报价高

技术对于不同的人的价格是完全不同的。技术的价格主要取决于买方。不同的买方就对应于不同的技术价格，技术的价格底线买方要开发出同样技术所必须付出的代价。2000 年行业寡头公司并购了中型公司，收购价格低于创业公司开发相应技术的十分之一，不包括销售的费用。低价的依据可能是寡头公司开发相应技术的成本。没有在软件公司上过班的人，很难对开发成本有概念。

初步估计，没有技术背景的建筑学创业者开发相应的技术，一般要花费 10 倍左右的成本。

收购技术获得的报价，大概也位于同一个区间。

3.7 B 端和 C 端的客户开发成本高于 VC 端

由于市场上 TO B, TO C 的竞争极为激烈，最近有个新趋势，竞争中失利的建筑学创业者转向了做寡头公司的全产业链建筑软件。仔细一看，用 `Three.js` 做的演示版。

能用的全产业链软件，必须用 C++ 等底层语言，开发核心图形库，要维持一个高水平的技术团队。代价非常大。耗时非常长。

也就是说，B 端和 C 端都受挫以后，做 VC 端的生意。